Big Data Analytics: Black Friday Merchandise Analytics and Prediction

Yuntong Wang, Woye Lin, Guowei Xu
Columbia University
E-mail: yw2768@columbia.edu, , wl2575@columbia.edu, gx2127@columbia.edu

Abstract—**E-commerce platforms and off-line retailers consider the Friday after Thanksgiving, Black Friday, as their most profitable holiday shopping day. Understanding consumers' attitudes and intentions towards shopping may aid retailers to perfect their holiday marketing strategies. This study uses purchases summary of various customers for selected high volume products from Black Friday month. Big data techniques are utilized to analyze the costumers' behaviors and make user-based recommendations. Retailer managers will have a better opportunity to promote on Black Friday with an understanding of consumer intentions for these major shopping occasions and advertise their products in a user-customized fashion.**

Black Friday; Big Data; Hadoop; Collaborate Filtering; Merchandise Analysis.

I. Introduction

The purpose of this study is to analyze the customers' behaviors and make user-based recommendations according to Black Friday Merchandise dataset. Black Friday, the day after Thanksgiving, is a term used by the retail industry in the United States that signifies the start of the Christmas holiday shopping season. As e-Commerce platforms like Amazon and eBay and off-line stores like Target, Walmart are concerned about how customers are shopping on various channels and categories, a deeper dive into purchase summaries during the peak holiday shopping season may provide a promising insight into shopping behaviors and habits. What are consumers' intentions during the hot shopping season, and how do they compare to previous shopping behaviors? How should retailers leverage their marketing strategies based on this useful data? To answer these questions, our objective is to research and understand consumer motivations and the factors behind this motivations, in order to provide needed answers to retailers in terms of how to base their holiday marketing budget. However, with large volume of purchase records during peak shopping season, traditional data analytics methods seem trivial. Thus we take the advantage of big data platforms like Hadoop and Spark, including software Hive and Mahout, to perform queries into dataset and machine learning algorithm to make recommendations in a user-based fashion. With this information, retailers can develop campaigns that target their audiences based on consumer channel and category preference for holiday shopping.

This study will mostly focus on three approaches of big data analysis, including basic data statistics using Spark, issuing queries to analyze customer behaviors using Hive with Hadoop and collaborative filtering recommendations with Mahout.

In order to solve scalability problem of huge dataset, we need to implement our system on top of cloud-computing platform. There're several cloud computing platforms available, for example, the DynamoDB of Amazon AWS [1], and Dryad of Microsoft [2]. For our study, we choose Hadoop platform as the base of our implementations. Since Hadoop is an open source cloud computing platform, it implements the MapReduce framework that have been successfully evaluated by Google [3]. Its distributed file system, Hadoop Distributed File System (HDFS), is well suited for distributed storage and distributed processing using commodity hardware. It is fault tolerant, scalable, and extremely simple to expand.

For performing data processing and analysis on high volume dataset, we use Hive, a fast real-time data warehousing solution as our database. Hive allows users to maintain and reuse custom queries and programs easily. SQL-like declarative language – HiveQL is supported in Hive, which is complied into map-reduce jobs executed on Hadoop [4].

In terms of implementation of Collaborative Filtering recommendations on cloud computing platform, we will utilize a very close related project, Mahout [5], which implements the Collaborative Filtering recommendation system base on Taste [6], a flexible, fast collaborative filtering engine for Java. It's an environment for quickly creating scalable performant machine learning applications.

II. Related Works

Previous works on merchandise analysis mainly fell into following categories:

First, from the perspective of behavioral economics commons [7] and psychological phenomenological interviewing [8], researchers observe and analyze the consumer behaviors and commentary of the Black Friday customer. The approaches taken include in-line and in-store observations, interviews and surveys. However, there will be ambiguity and inconsistency between real purchase data and interview in person. And big data of high volume purchase records, analysis of database and statistic information processing in mathematical way are not applied in these cases.

Second category of related works proves the usefulness and potentials of big data analytics in the field of business intelligence and analytics, instead of providing reusable system design and real data analysis. With the ultra-fast global IT connections, the development and deployment of business-related data standards, electronic data interchange (EDI) formats, and business databases and information systems have greatly facilitated business data creation, analysis and utilization [9]. Our work focuses on offering a promising solution that utilizes the real-world semi-structured data, calling for ad hoc and one-time extraction, parsing, processing, indexing, and analytics in a scalable and distributed MapReduce and Hadoop environment. Other related works approaches the problem with different algorithms, like data mining algorithms instead of machine learning. Meanwhile, some of them focuses on a relatively broad dataset other than peak holiday shopping season like Black Friday [10]. However, in this study, machine learning algorithm is specifically applied to the dataset of the most popular shopping event in U.S., which presents a more direct and robust marketing aid to retailers.

III. System Overview

The system is described in Figure 1. We first download the structured dataset, which is stated detailed below, then prepare csv staging files into HDFS on Hadoop platform. After proper data import, we use Hive, Mahout and IBM System G to process and analyze the data separately.
The dataset we used in this study is from a data analytics website named analyticsvidhya.com. The data includes millions of records of the purchases of various customers for selected high volume products from Black Friday month provided by an on-line retailer, providing enough resources for us to perform analytical queries and collaborative filtering on.
The data set also contains customer demographics (age, gender, marital status, city_type, stay_in_current_city), product details (product_id and product category) and Total purchase_amount from last month. The full description of data structure is described below.

Variable	Definition
User_ID	User ID
Product_ID	Product ID
Gender	Sex of User
Age	Age in bins
Occupation	Occupation
City_Category	Category of the City
Stay In Current City Years	Number of years stay in current city
Marital_Status	Marital Status
Product Category 1	Product Category
Product Category 2	Product may belongs to other category also
Product Category 3	Product may belongs to other category also (Masked)
Purchase	Purchase Amount (Target Variable)

Table 1 Data Description

IV. Algorithm

We use collaborative filtering algorithm to implement a product recommender. One advantage of the collaborative filtering algorithm is that it doesn't require any feature about user or item. It only takes as input the "User", "Item", and preference between a "User" and an "Item".

To do collaborative filtering on our dataset, we first write a python script to process the data by filtering out unnecessary columns and only leave "user_id", "product_id", and "purchase_amount". "User_id" and "product_id" refer to

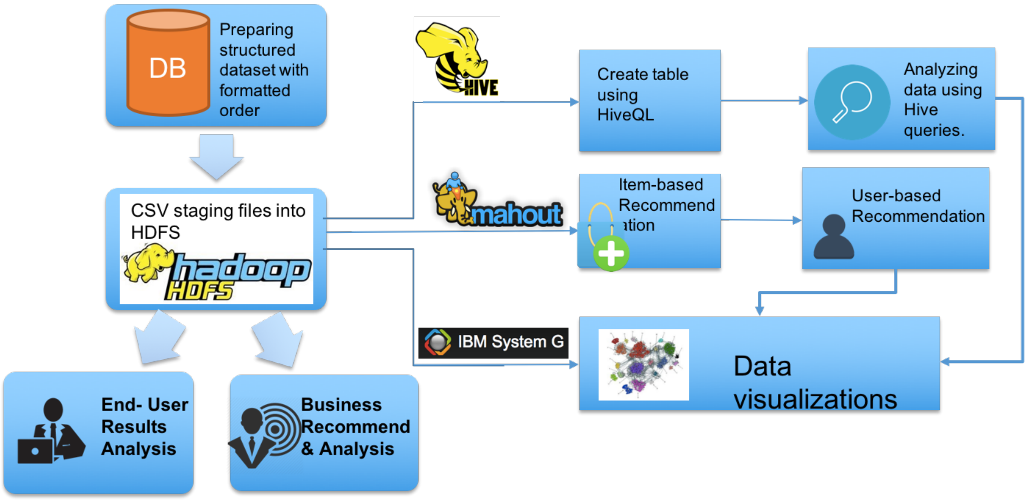

DB
Preparing structured dataset with formatted order
CSV staging files into HDFS
hadoop HDFS
HIVE
Create table using HiveQL
Analyzing data using Hive queries.
mahout
Item-based Recommend ation
User-based Recommendation
IBM System G
Data visualizations
End- User Results Analysis
Business Recommend & Analysis

Figure 1 System Flowchart

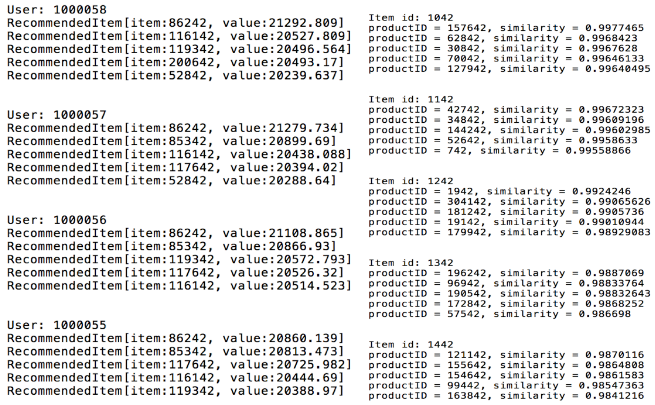
users and products respectively. Typically, the preference is represented by user's rating of the product; however our dataset does not contain such rating. After careful consideration, we believe that the "Purchase_amount" can be used as user's preference to a product. We justify this by following reasons:

- We believe that a high "purchase_amount" indicates that the user really loves a product and such purchased this product frequently, which is a strong signal that the user has a strong preference for this product

- It's also possible that unit price of different products may cause the purchase amount vary from each other. We consider this concern and justify the use of purchase_amount as the intersection strength measure by the analysis that certain group of users always tends to, or rich enough to be able to buy expensive stuffs, while others may prefer cheap products. So for the first group of users, we will focus on recommending expensive products, while for the second group of users, we will recommend cheap products.

- Also we believe that if a user choose to buy a product even though it is very expensive, then it must be the case that the user have a strong preference for the product, because people usually very cautious when they decide to purchase an expensive product, and they want to make sure that they spend such large amount of money on the most favorible product.

For above reasons, we believe that it is valid to use purchase amount as a measure of user's preference to a product.
We create a maven project to implement our collaborative filter. Below are our recommender's sample outputs:



```
User: 1000058
RecommendedItem[item:86242, value:21292.809]
RecommendedItem[item:116142, value:20527.809]
RecommendedItem[item:119342, value:20496.564]
RecommendedItem[item:200642, value:20493.17]
RecommendedItem[item:52842, value:20239.637]


User: 1000057
RecommendedItem[item:86242, value:21279.734]
RecommendedItem[item:85342, value:20899.69]
RecommendedItem[item:116142, value:20438.088]
RecommendedItem[item:117642, value:20394.02]
RecommendedItem[item:52842, value:20288.64]


User: 1000056
RecommendedItem[item:86242, value:21108.865]
RecommendedItem[item:85342, value:20866.93]
RecommendedItem[item:119342, value:20572.793]
RecommendedItem[item:117642, value:20526.32]
RecommendedItem[item:116142, value:20514.523]


User: 1000055
RecommendedItem[item:86242, value:20860.139]
RecommendedItem[item:85342, value:20813.473]
RecommendedItem[item:117642, value:20725.982]
RecommendedItem[item:116142, value:20444.69]
RecommendedItem[item:119342, value:20388.97]
```

```
Item id: 1042
productID = 157642, similarity = 0.9977465
productID = 62842, similarity = 0.9968423
productID = 30842, similarity = 0.9967628
productID = 70042, similarity = 0.99646133
productID = 127942, similarity = 0.99640495

Item id: 1142
productID = 42742, similarity = 0.99672323
productID = 34842, similarity = 0.99609196
productID = 144242, similarity = 0.99602985
productID = 52642, similarity = 0.9958633
productID = 742, similarity = 0.99558866

Item id: 1242
productID = 1942, similarity = 0.9924246
productID = 304142, similarity = 0.99065626
productID = 181242, similarity = 0.9905736
productID = 19142, similarity = 0.99010944
productID = 179942, similarity = 0.98929083

Item id: 1342
productID = 196242, similarity = 0.9887069
productID = 96942, similarity = 0.98833764
productID = 190542, similarity = 0.98832643
productID = 172842, similarity = 0.9868252
productID = 57542, similarity = 0.986698

Item id: 1442
productID = 121142, similarity = 0.9870116
productID = 155642, similarity = 0.9864808
productID = 154642, similarity = 0.9861583
productID = 99442, similarity = 0.98547363
productID = 163842, similarity = 0.9841216
```

Figure 2 Recommender's Sample Outputs

We write our collaborative filter based on mahout engine as follows:

User-based Recommender:
User-based recommenders are the typical conventional style of a recommender.

- Create a DataModel from the processed file that contains only "UID", "PID", "Purchase_Amount"

- Extract user similarity based on "PearsonCorrelationSimilarity"

- Create "UserNeighborhood" on top of previous created "userSimilarity" and "Model".

- Create the recommender "GenericBasedUserRecommender"

- Loop through all the users and apply the recommender to actually recommend 5 products to each of the user.

Item-based recommender:
Unlike user-based recommender, item-based recommender base recommendation on item-similarity. Generally speaking, it takes an item as input and find out the similar items that are most similar to this item. For large data sets, item-based recommenders are more appropriate.

- Create DataModel based on filtered dataset.

- Extract itemSimilarity using "PearsonCorrelationSimilarity"

- Create the final recommender based on the "similarity" and "neighborhood"

- Loop through all the items and apply the recommender on each of the item to recommend 5 most similar products.

V. SOFTWARE PACKAGE DESCRIPTION

Mahout-based Collaborative filtering algorithm:

To pre-process our dataset to filter out the unecessary columns, we use following softwares:

- Python 2.7: Python is a programming language that lets you work quickly and integrate systems more effectively. Version 2.7 is scheduled to be the last major version in the 2.x series before it moves into an extended maintenance period.

- Pandas: Pandas is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language.

- Numpy: NumPy is the fundamental package for scientific computing with Python

To implement collaborative filtering recommender, we use the following platform and programming language:

- Eclipse J2EE Package: Tools for Java developers creating Java EE and Web applications, including a Java IDE, tools for Java EE, JPA, JSF, Mylyn, EGit and others. We use "Eclipse Java EE developer tools" and "Maven Integration for Eclipse"

- Mahout-based collaborative filtering engine, including packages such as "org.apache.mahout.cf", "com.predictionmarketing.Recommender"

Visualization software packages:
ggplot2 is a data visualization package for the statistical programming language R. Created by Hadley Wickham in 2005, ggplot2 is an implementation of Leland Wilkinson's Grammar of Graphics—a general scheme for data visualization which breaks up graphs into semantic components such as scales and layers. ggplot2 can serve as a replacement for the base graphics in R and contains a number of defaults for web and print display of common scales.[11]

VI. Experiment Results

General data analysis using Spark and R:
First we want to know if people in different cities have different purchase level. We did a query in Spark and plot the outcome using ggplot2 package in R. As seen below, city A has a slightly lower purchase amount in various purchase levels compared to city B and city C, which are pretty close to each other.

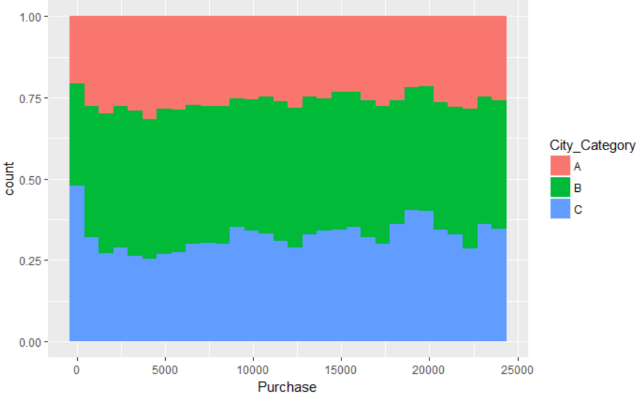


Figure 3 Purchase distribution in cities

Similarly, we did similar visualization among different genders, occupations and age groups.

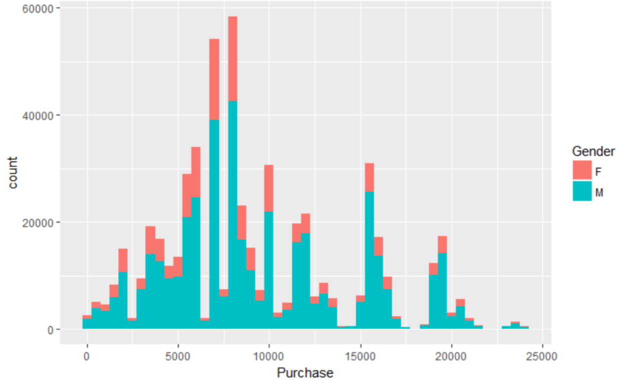


Figure 4 Purchase distribution in Males and Females

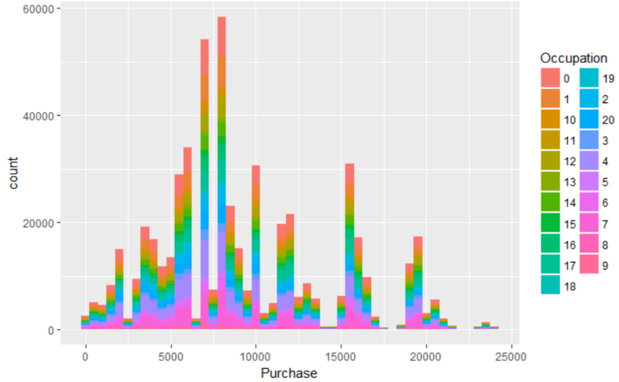


Figure 5 Purchase distribution in different occupations

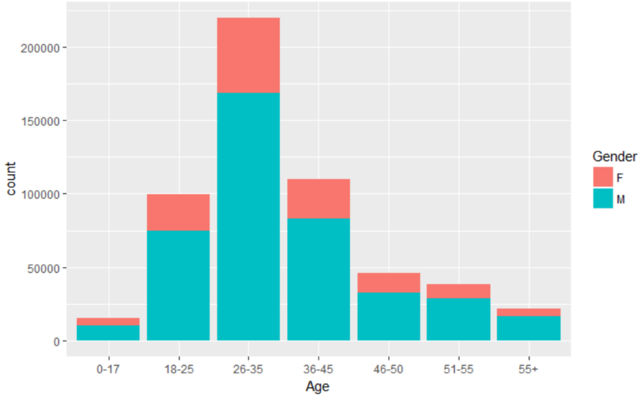


Figure 6 Purchase distribution among age groups

Our findings:

1. Females account for a small part of the records in all purchase levels. The exact figures are 414,259 (75.3%) records for males and 135,809 (24.7%) records for females. Therefore, conclusions drew from this dataset mostly apply to males' purchase feature.
2. People in age group 26-35 are the biggest part of all costumers. Sellers should really focus on this group.

3. The purchase amount ranges from $0 to around $23,000 with the bulk falls within $5,000 to $10,000. Sellers should also focus on people who have large purchase amount over $17,000.

The next thing we did is to divide all customers into 4 tiers based on their purchase amounts:
“Tier1”: over $17,500
“Tier2”: from $10,000 to $17,500
“Tier3”: from $5,000 to $10,000
“Tier4”: below $5,000
Since “age” and “purchase amount tier” are two important factors from previous analysis, we want to see more of their relationships.

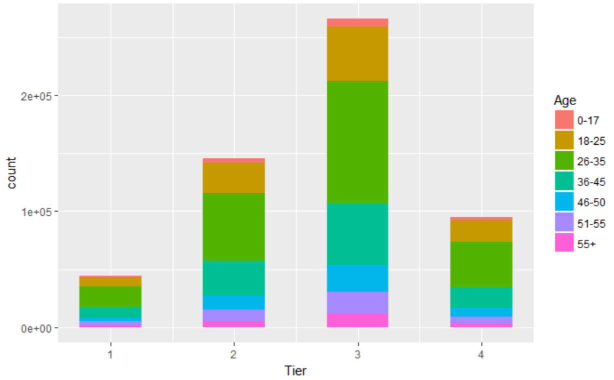


Figure 7: Age group distribution in different tiers

This plot indicates that age groups actually have similar patterns in 4 tiers, people from 26-35 should always be the biggest potential buyers.

Our next question is “do people in different tiers purchase one kind of products more than the others?” We plot Pi charts for average number of products from product category 1 to category 3.

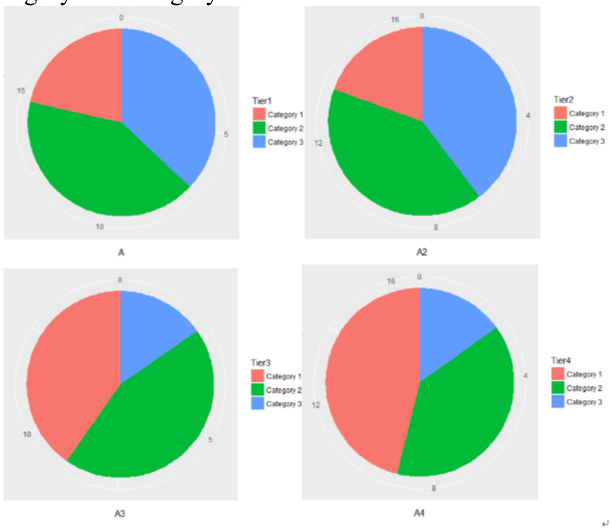


Figure 8: Product categories in different tiers

Our finding:
Interestingly for tier 1 and tier 2 who have large purchase amounts, category 2 and 3 have a close share while category 1’s share is obviously less. However in tier 3 and tier4, category 1 has the largest share followed by category 2 and then category 3.

Accordingly, sellers should recommend more category 1 and 2 items to buyers with high purchase amounts but more category 1 items to people with low purchase amounts.

Lastly, we want to find out what are the most popular “Black Friday” products among groups?

Group	Top1	Top2	Top3	Top4	Top5
Tier1	P0002544 2	P0011094 2	P0008034 2	P0005284 2	P0011074 2
Tier2	P0014504 2	P0004674 2	P0011214 2	P0000014 2	P0005764 2
Tier3	P0026524 2	P0011794 2	P0005804 2	P0022044 2	P0011744 2
Tier4	P0010264 2	P0000344 2	P0037164 4	P0037244 5	P0037085 3
Age25 -36	P0026524 2	P0011074 2	P0002544 2	P0011214 2	P0005764 2
Total	P0026524 2	P0002544 2	P0011074 2	P0011214 2	P0005764 2

Table 2 the most popular “Black Friday” products among groups

We found out that item P00265242 is the most purchased product this Black Friday.

CF filtering results

In additiona to the sample output we shows in the previous sections, we also would like to evaluate our recommender to make sure that the recommended items are effective. We do the evaluation using statistical offline evaluation. The way we do that is to take a hold-out test, partitioning our dataset into training data and test data, in a proportion of 9:1. We train our recommender using the training data, and evaluate the recommender using the test data.

We implement this by creating a class “EvaluateRecommender” and add an inner class “MyRecommenderBuilder”, which implements the “RecommenderBuilder interface. To test our recommender, we check how much the recommender misses the real interaction strength by employing “AverageAbsoluteDifferenceRecommenderEvaluator”. Our result is 1935.794629490881. We calculate the average interaction strength of our dataset, which is 19263.96871296. Therefore, the average strength lost from our recommender is: 1935.794629490881 / 19263.96871296= 10.04%. We believe that this is an acceptable lost rate.

VII. Conclusion

Basic Spark results summary:

1. All conclusions drew from this dataset should mostly apply to male buyers since male records account for almost 75% of all records.
2. Age group 25-36 should be the biggest potential buyers for all product categories.
3. People who have purchased a lot prefer to buy category 2 and 3 products than category 1 products; people with lower purchase amounts prefer category 1 items.
4. Product P00265242 is the best selling item.

Last name	Xu	Lin	Wang
Contribution	33.3%	33.3%	33.3%
Workload 1	Basic data analysis using Spark	Pre-process dataset for collaborative filtering algorithm	Find suitable dataset for the project
Workload 2	Data visualization	Implement User-based, Item-based recommender	Analyze data by issuing queries in Hive
Workload 3	Summarize and show results of Spark and data visualization work in report	Evaluate the recommender by implementing recommender evaluator	Summarize our work and compare with related works in first three part of the report.

Table 3 Workload Division

Appendix

References

[1] DeCandia, Giuseppe, et al. "Dynamo: amazon's highly available key-value store." ACM SIGOPS Operating Systems Review 41.6 (2007): 205-220.

[2] Isard, Michael, et al. "Dryad: distributed data-parallel programs from sequential building blocks." *ACM SIGOPS Operating Systems Review*. Vol. 41. No. 3. ACM, 2007.

[3] Dean, Jeffrey, and Sanjay Ghemawat. "MapReduce: simplified data processing on large clusters." *Communications of the ACM* 51.1 (2008): 107-113.

[4] Thusoo, Ashish, et al. "Hive: a warehousing solution over a map-reduce framework." *Proceedings of the VLDB Endowment* 2.2 (2009): 1626-1629.

[5] http://lucene.apache.org/mahout/

[6] http://taste.sourceforge.net/

[7] Simpson, Linda, et al. "An analysis of consumer behavior on Black Friday." *American International Journal of Contemporary Research* (2011).

[8] Boyd Thomas, Jane, and Cara Peters. "An exploratory investigation of Black Friday consumption rituals." *International Journal of Retail & Distribution Management* 39.7 (2011): 522-537.

[9] Chen, Hsinchun, Roger HL Chiang, and Veda C. Storey. "Business Intelligence and Analytics: From Big Data to Big Impact." *MIS quarterly* 36.4 (2012): 1165-1188.

[10] Ling, Charles X., and Chenghui Li. "Data Mining for Direct Marketing: Problems and Solutions." *KDD*. Vol. 98. 1998.

[11] https://en.wikipedia.org/wiki/Ggplot2

[12] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. *(references)*

[13] I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[14] K. Elissa, "Title of paper if known," unpublished.

[15] R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.

[16] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[17] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

[18] Electronic Publication: Digital Object Identifiers (DOIs):Article in a journal:

[19] D. Kornack and P. Rakic, "Cell Proliferation without Neurogenesis in Adult Primate Neocortex," Science, vol. 294, Dec. 2001, pp. 2127-2130, doi:10.1126/science.1065467.

Article in a conference proceedings:

[20] H. Goto, Y. Hasegawa, and M. Tanaka, "Efficient Scheduling Focusing on the Duality of MPL Representatives," Proc. IEEE Symp. Computational Intelligence in Scheduling (SCIS 07), IEEE Press, Dec. 2007, pp. 57-64, doi:10.1109/SCIS.2007.357670.

[21] B. Sarwar, G. Karypis, J. Konstan and J. Riedl, "Item-based collaborative filtering recommendation algorithms " in Proceedings of the Tenth International Conference on the World Wide Web (WWW 10), pp. 285-295, 2001.

[22] J.L. Herlocker, J.A. Konstan, A. Borchers and J. Riedl, "An algorithmic framework for performing collaborative filtering " in Proceedings of the 22nd annual international ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 99), pp. 230-237, 1999.